International Journal of Foundations of Computer Science
Vol. 30, Nos. 6 & 7 (2019) 1177–1196

DOI: 10.1142/S0129054119400343





# Reset Complexity of Ideal Languages Over a Binary Alphabet

Marina Maslennikova

*Institute of Natural Sciences and Mathematics*
*Ural Federal University, 620000 Ekaterinburg, Russia*
*maslennikova.marina@gmail.com*

Received 16 November 2017
Accepted 5 June 2018
Communicated by C. Câmpeanu and G. Pighizzini

We prove **PSPACE**-completeness of checking whether a given ideal language serves as the language of reset words for some automaton with at most four states over a binary alphabet. We compare the reset complexity and the state complexity for languages related to slowly synchronizing automata.

*Keywords*: Ideal language; synchronizing automaton; reset word; reset complexity; state complexity; **PSPACE**-completeness.

## 1. Introduction

Regular languages admit compact representations by different tools: deterministic and nondeterministic finite automata, syntactic monoids, regular expressions, etc. Each of these tools gives rise to the corresponding complexity measure of regular languages. Along with these general tools, there are more specific devices for representing regular languages from some special classes. One of such classes is formed by ideal regular languages. A language $I \subseteq \Sigma^*$ is called a *two-sided ideal* (or simply an *ideal*) if $I$ is non-empty and $\Sigma^* I \Sigma^* \subseteq I$. In what follows we consider only languages which are regular, thus we drop the adjective "regular". Thus, the expression "ideal language" (or simply "ideal") always means a regular two-sided ideal language.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a *deterministic finite automaton* (DFA), where $Q$ is the *state set*, $\Sigma$ stands for the *input alphabet*, and $\delta\colon Q \times \Sigma \to Q$ is the totally defined *transition function* defining the action of the letters in $\Sigma$ on $Q$. The function $\delta$ is extended uniquely to a function $Q \times \Sigma^* \to Q$, where $\Sigma^*$ stands for the free monoid over $\Sigma$. The latter function is still denoted by $\delta$. In the theory of formal languages the definition of a DFA usually includes the *initial state* $q_0 \in Q$ and the set $F \subseteq Q$ of *terminal states*. We use these ingredients when dealing with automata as devices for recognizing languages. A language $L \subseteq \Sigma^*$ is *recognized* by an automaton $\mathscr{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ if $L = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. We denote by $L[\mathscr{A}]$ the language recognized by the automaton $\mathscr{A}$. We also use standard concepts

of the theory of formal languages and computational complexity theory such as regular language, minimal automaton etc. [12].

A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is called *synchronizing* if there exists a word $w \in \Sigma^*$ whose action leaves the automaton in one particular state no matter at which state in $Q$ it is applied: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$. Any word $w$ with this property is said to be *reset* for the DFA $\mathscr{A}$. The minimum length of reset words for $\mathscr{A}$ is called the *reset threshold* of $\mathscr{A}$ and is denoted by $rt(\mathscr{A})$. For the last 50 years synchronizing automata have received a great deal of attention. For a brief introduction to the theory of synchronizing automata we refer the reader to the survey [14].

In the present paper we focus on some complexity aspects of the theory of synchronizing automata. We denote by $\mathrm{Syn}(\mathscr{A})$ the language of reset words for a given automaton $\mathscr{A}$. It is well known that $\mathrm{Syn}(\mathscr{A})$ is regular [14]. Furthermore, it is an ideal in $\Sigma^*$. On the other hand, every regular ideal language $I$ serves as the language of reset words for some automaton. For instance, the minimal automaton recognizing $I$ is synchronized exactly by words from $I$ [8]. Thus, synchronizing automata can be considered as a special representation of ideal languages. Effectiveness of such a representation was addressed in Ref. [8]. The *reset complexity* $rc(I)$ of an ideal language $I$ is the minimal possible number of states in a synchronizing automaton $\mathscr{A}$ such that $\mathrm{Syn}(\mathscr{A}) = I$. Every such automaton $\mathscr{A}$ is called a *minimal synchronizing automaton* (for brevity, MSA).

From descriptive complexity point of view it is interesting to compare the reset complexity of an ideal language with the classical state complexity. The *state complexity* $sc(I)$ of a regular language $I$ is the number of states in the minimal automaton $\mathscr{A}_I$ recognizing $I$. For every ideal language $I$, we have $rc(I) \leq sc(I)$ [8]. Moreover, for each $n \geq 3$, there exists a language $I_n$ such that $rc(I_n) = n$ and $sc(I_n) = 2^n - n$ [8]. Therefore, the representation of an ideal language by means of a synchronizing automaton can be exponentially more succinct than the "traditional" representation via the minimal automaton. This resembles the well-known property of nondeterministic finite automata (for brevity, NFAs): for each $n \geq 3$, there is an $n$-state NFA $\mathscr{N}$ such that every DFA recognizing the same language as $\mathscr{N}$ has exactly $2^n$ states [10, 11].

Another source of motivation for studying representations of ideal languages by means of synchronizing automata comes from the famous Černý's conjecture [3]. Černý conjectured that every synchronizing automaton with $n$ states possesses a reset word of length at most $(n-1)^2$. Let $||I||$ be the minimal length of words in $I$. The Černý conjecture holds true if and only if $rc(I) \geq \sqrt{||I||} + 1$ for every ideal $I$. The latter inequality would provide the desired quadratic upper bound on the reset threshold of a synchronizing automaton. Even a lower bound $rc(I) \geq \frac{\sqrt{||I||}}{C}$ for some constant $C$ would be a major breakthrough.

Let $I$ be an ideal regular language over $\Sigma$ with $rc(I) = n$. The latter equality means that there exists some $n$-state DFA $\mathscr{B}$ such that $\mathrm{Syn}(\mathscr{B}) = I$, and $\mathscr{B}$ is an MSA for $I$. The following question arises: how hard is it to check that a given

synchronizing DFA $\mathscr{B}$ is an MSA for a given ideal $I$ ($I$ is assumed to be given by a synchronizing DFA $\mathscr{A}$ with $\mathrm{Syn}(\mathscr{A}) = I$)? Another question related to the previous one is the following: how hard is it to verify the inequality $rc(I) \leq \ell$ for a given ideal $I$ and a given $\ell \in \mathbb{N}$? The inequality $rc(I) \leq \ell$ means that there exists a synchronizing DFA $\mathscr{B}$ with at most $\ell$ states such that $\mathrm{Syn}(\mathscr{B}) = I$. The aforementioned questions are trivial for automata over a unary alphabet, thus in what follows we deal with alphabets that have at least two letters. The problem of checking the equality $\mathrm{Syn}(\mathscr{B}) = I$ is equivalent to the problem of checking the equality $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$ for two given synchronizing automata $\mathscr{A}$ and $\mathscr{B}$. The complexity of the latter problem has been partially studied in Ref. [9]. It is well known that the equality of the languages recognized by two given DFAs can be checked in time polynomial of the size of automata. However, the problem of checking the equality of the languages of reset words of two synchronizing DFAs turns out to be **PSPACE**-complete [9]. Recall that the problem of checking the equality of languages recognized by two given NFAs is **PSPACE**-complete as well [13]. In this context we again find that synchronizing automata share some properties of nondeterministic finite automata.

We state formally the SYN-EQUALITY problem:

– *Input:* synchronizing automata $\mathscr{A}$ and $\mathscr{B}$.
– *Question:* is $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$?

In Ref. [9] SYN-EQUALITY has been proved to be **PSPACE**-complete. In the present paper we provide a more transparent proof of **PSPACE**-hardness of this problem. In particular, it allows us to strengthen the result of [9] concerning the problem of evaluating the reset complexity of a given ideal language.

We state formally the RESET-INEQUALITY problem:

– *Input:* synchronizing DFA $\mathscr{A}$ over $\Sigma$, $\ell \in \mathbb{N}$.
– *Question:* is $rc(\mathrm{Syn}(\mathscr{A})) \leq \ell$?

In Ref. [9] RESET-INEQUALITY has been shown to be **PSPACE**-complete for $\ell = 3$ and enough large input alphabet (with at least five letters). In the present paper we significantly strengthen this result and prove that RESET-INEQUALITY, restricted to a binary alphabet, remains **PSPACE**-complete even for $\ell = 4$. Note that RESET-INEQUALITY is trivial for DFAs over a unary alphabet and, furthermore, RESET-INEQUALITY can be solved in polynomial of the size of $\mathscr{A}$ time for $\ell = 1$ and $\ell = 2$ in the general case [9]. So the only question that remains open concerns the complexity of RESET-INEQUALITY for $\ell = 3$ for the binary alphabet case.

Further we study ideal languages $I_n$ which are certificates for exponential gap between the state complexity and the reset complexity, that is $rc(I_n) = n$ and $sc(I_n) = 2^n - n$. One of such examples has been presented in Ref. [8] and, actually, it is provided by the construction of Černý's automaton $\mathscr{C}_n$. Namely, it is taken

as $I_n = \mathrm{Syn}(\mathscr{C}_n)$. Let us note that the DFA $\mathscr{C}_n$ is one of well-known examples of "slowly" synchronizing automata, that is automata whose reset threshold is close to $(n-1)^2$. The following question arises: does slow synchronization of a given DFA $\mathscr{A}_n$ guarantee that the state complexity of $\mathrm{Syn}(\mathscr{A}_n)$ is exponentially smaller than the reset complexity of this language? We study several series of slowly synchronizing automata known from [1] and prove that the considered automata are certificates for an exponential gap between the state complexity and the reset complexity of the corresponding ideal language. Further it would be interesting to calculate another complexity measures of languages of reset words for such automata (nondeterministic state complexity, syntactic complexity, etc.) and compare them with known values of reset and state complexity.

The paper is organized as follows. In Sec. 2, we introduce some definitions and preliminary results. In Sec. 3, we provide a modified proof of **PSPACE**-hardness of SYN-EQUALITY. Section 4 contains the main result about **PSPACE**-completeness of the problem RESET-INEQUALITY for the binary alphabet case. In Sec. 5, we compare the state complexity and the reset complexity of languages related to slowly synchronizing automata.

## 2. Preliminaries

A standard tool for finding the language of reset words of a given DFA $\mathscr{K} = \langle Q, \delta, \Sigma\rangle$ is the *power automaton* $\mathcal{P}(\mathscr{K})$. Its state set is the set $\mathcal{Q}$ of all nonempty subsets of $Q$, and the transition function is defined as a natural extension of $\delta$ to the set $\mathcal{Q}\times\Sigma$ (the resulting function is also denoted by $\delta$), namely, $\delta(S, c) = \{\delta(q, c) \mid q \in S\}$ for $S \subseteq Q$ and $c \in \Sigma$. If we take the set $Q$ as the initial state and singletons as final states in $\mathcal{P}(\mathscr{K})$, then we obtain an automaton recognizing $\mathrm{Syn}(\mathscr{K})$. It is easy to see that if all singletons are merged into a unique sink state $s$, the resulting automaton still recognizes $\mathrm{Syn}(\mathscr{K})$. Throughout the paper the term *power automaton* and the notation $\mathcal{P}(\mathscr{K})$ refer to this modified version.

A state $s$ of a DFA $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ is said to be a *sink* if $\delta(s, a) = s$ for all $a \in \Sigma$. If the transition function $\delta$ is clear from the context, we write $q\,.\,w$ instead of $\delta(q, w)$ for $q \in Q$ and $w \in \Sigma^*$. This notation extends naturally to any subset $H \subseteq Q$ by putting $H\,.\,w = \{\delta(q, w) \mid q \in H\}$.

Recall that a word $u \in \Sigma^*$ is a *prefix* (*suffix* or *factor*, respectively) of a word $w$ if $w = us$ ($w = tu$ or $w = tus$, respectively) for some $t, s \in \Sigma^*$. A reset word for a DFA $\mathscr{A}$ is called *minimal* if none of its proper prefixes nor suffixes is reset.

Due to [9, Corollary 1] we have the following proposition.

**Proposition 1.** *SYN-EQUALITY is in* **PSPACE**.

To prove that SYN-EQUALITY is a **PSPACE**-complete problem we reduce the following well-known **PSPACE**-complete problem to the complement of SYN-EQUALITY. This problem deals with checking emptiness of the intersection of languages recognized by DFAs from a given collection [6].

FINITE AUTOMATA INTERSECTION

– *Input:* an integer $n$ and $n$ DFAs $M_i = \langle Q_i, \Sigma, \delta_i, q_i, F_i\rangle$, for $i = 1, \ldots, n$.
– *Question:* is $\bigcap_i L[M_i] \neq \varnothing$?

Since FINITE AUTOMATA INTERSECTION is known to be **PSPACE**-complete even for the binary alphabet case, we may assume that $|\Sigma| = 2$, in particular, let $\Sigma$ be $\{a, b\}$.



## 3. PSPACE-Hardness of SYN-EQUALITY

The proof of **PSPACE**-hardness of SYN-EQUALITY is based on the same idea as the proof from [9]. However, our modified construction allows us to reduce the number of letters from 5 to 4. We provide a sketch of the proof of **PSPACE**-hardness for the sake of completeness.

Given an instance of FINITE AUTOMATA INTERSECTION, we can assume without loss of generality that each initial state $q_i$ has no incoming edges and $q_i \notin F_i$. Indeed, excluding the case for which the empty word $\varepsilon$ is in $L[M_i]$ we can always build a DFA $M_i' = \langle Q_i', \Sigma, \delta_i', q_i', F_i\rangle$, which recognizes the same language as $M_i$, such that the initial state $q_i'$ has no incoming edges. This can easily be achieved by adding a new initial state $q_i'$ to the state set $Q_i$ and defining the transition function $\delta_i'$ by the rule: $\delta_i'(q_i', c) = \delta_i(q_i, c)$ for all $c \in \Sigma$ and $\delta_i'(q, c) = \delta_i(q, c)$ for all $c \in \Sigma,\ q \in Q_i$. Furthermore, we may assume that the sets $Q_i$, for $i = 1, \ldots, n$, are pairwise disjoint. Also we can suppose that any letter from $\Sigma$ does not belong to $\bigcap_i L[M_i]$. Otherwise, we add a new initial state $q_i''$ to each $M_i'$ and put $\delta_i(q_i'', c) = q_i'$ for all $c \in \Sigma$. This assumption will be of use in Sec. 3.

To build an instance of SYN-EQUALITY from the DFAs $M_i$, $i = 1, \ldots, n$, we construct a DFA $\mathscr{A} = \langle Q, \Delta, \varphi\rangle$ with $Q = \bigcup_{i=1}^{n} Q_i \cup \{s, h\}$, where $s$ and $h$ are new states not belonging to any $Q_i$. We add two new letters $x$ and $z$ to the alphabet $\Sigma$ and let $\Delta = \Sigma \cup \{x, z\}$. The transition function $\varphi$ of the DFA $\mathscr{A}$ is defined by the following rules:

$$\begin{aligned}
&\varphi(q, c) = \delta_i(q, c) && \text{for all } i = 1, \ldots, n,\ q \in Q_i \text{ and } c \in \Sigma;\\
&\varphi(q, x) = q_i && \text{for all } i = 1, \ldots, n,\ q \in Q_i;\\
&\varphi(q, z) = s && \text{for all } i = 1, \ldots, n,\ q \in F_i;\\
&\varphi(q, z) = h && \text{for all } i = 1, \ldots, n,\ q \in Q_i \backslash F_i;\\
&\varphi(h, c) = s && \text{for all } c \in \Delta;\\
&\varphi(s, c) = s && \text{for all } c \in \Delta.
\end{aligned}$$

The resulting automaton $\mathscr{A}$ is shown schematically in Fig. 1. The action of letters from $\Sigma$ on the states $p \in Q_i$ is not shown. Denote by $G_i$ the set $Q_i \backslash (F_i \cup \{q_i\})$. All the states from the set $G_i$ are shown as the node labeled by $G_i$. All the states from the set $F_i$ are shown as the node labeled by $F_i$.

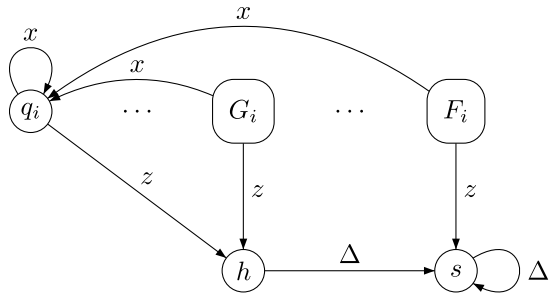


Fig. 1. Automaton $\mathscr{A}$.

The constructed automaton is synchronizing, for example, by the word $zz$. It can be easily seen that by the definition of the transition function $\varphi$ we get $\varphi(Q, w) \cap Q_i \neq \varnothing$ if and only if $w \in (\Sigma \cup \{x\})^*$. Consider the language

$$I = (\Sigma \cup \{x\})^* z \Delta^+.$$

From the observations above and the definition of $\varphi$ we obtain Lemma 2.

**Lemma 2.** $\bigcap_{i=1}^n L[M_i] = \varnothing$ *if and only if* $\mathrm{Syn}(\mathscr{A}) = I$.

**Proof.** Let $\bigcap_{i=1}^n L[M_i] = \varnothing$. It is easy to check that if $\bigcap_{i=1}^n L[M_i] = \varnothing$, then $\mathrm{Syn}(\mathscr{A}) \subseteq I$. The opposite inclusion $I \subseteq \mathrm{Syn}(\mathscr{A})$ follows easily from the construction of $\mathscr{A}$. Assume now that the equality $\mathrm{Syn}(\mathscr{A}) = I$ takes place, and $\bigcap_{i=1}^n L[M_i] \neq \varnothing$, thus there is some $w' \in \bigcap_{i=1}^n L[M_i]$. By the definition of $\varphi$ we get that the word $w = xw'z$ is a reset word for $\mathscr{A}$. However, $w \notin I$. So we come to a contradiction. □

Now we build a 3-state automaton $\mathscr{B} = \langle P, \Delta, \tau \rangle$ (see Fig. 2). Its state set is $P = \{p_1, p_2, s'\}$, where $s'$ is a unique sink state. It is easy to verify that $I$ serves as the language of reset words for $\mathscr{B}$. Furthermore, $I$ does not serve as the language of reset words for a synchronizing automaton of size at most two over the same alphabet $\Delta$. So $\mathscr{B}$ is an MSA for $I$ and $rc(\mathrm{Syn}(\mathscr{B})) = 3$.

**Lemma 3.** $\mathrm{Syn}(\mathscr{B}) = I$.

Finally, by Lemmas 2 and 3, we have the following claim.

**Corollary 4.** $\bigcap_{i=1}^n L[M_i] = \varnothing$ *if and only if* $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$.

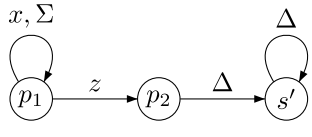


Fig. 2. Automaton $\mathscr{B}$.

## 4. PSPACE-Completeness of RESET-INEQUALITY

We have reduced the problem FINITE AUTOMATA INTERSECTION to the complement of SYN-EQUALITY. By construction of DFAs $\mathscr{A}$ and $\mathscr{B}$, we have $\Delta = \{z, a, b, x\}$. Now we are going to study the complexity of checking the inequality $rc(I) \leq \ell$ for the binary alphabet case. First we build DFAs $\mathscr{C} = \langle C, \{\mu, \lambda\}, \varphi_2\rangle$ and $\mathscr{D} = \langle D, \{\mu, \lambda\}, \tau_2\rangle$ over a binary alphabet $\{\mu, \lambda\}$ with unique sink states $\zeta_1$ and $\zeta_2$ respectively. It will turn out that the constructions of $\mathscr{C}$ and $\mathscr{D}$ preserve the equality of reset languages. More precisely, $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$ if and only if $\mathrm{Syn}(\mathscr{C}) = \mathrm{Syn}(\mathscr{D})$. Let $J = \mathrm{Syn}(\mathscr{C})$. We will prove that $rc(J) > 4$ if and only if $\bigcap_{i=1}^{n} L[M_i] \neq \varnothing$. It allows to obtain the desired result about **PSPACE**-completeness of RESET-INEQUALITY for the binary alphabet case.

In order to construct $\mathscr{C} = \langle C, \{\mu, \lambda\}, \varphi_2\rangle$ and $\mathscr{D} = \langle D, \{\mu, \lambda\}, \tau_2\rangle$ we apply a recoding technique which has been used in Refs. [1, 7, 9]. Namely, we define morphisms $h\colon \{\lambda, \mu\}^*\lambda \to \Delta^*$ and $\overline{h}\colon \Delta^* \to \{\lambda, \mu\}^*\lambda$ preserving the property of being a reset word for the corresponding automaton. Since the definitions of morphisms differ from those described in Ref. [9], we present them here. Let $d_1 = z$, $d_2 = a$, $d_3 = b$ and $d_4 = x$. We put $h(\mu^k\lambda) = d_{k+1}$ for $k = 0, \ldots, 3$ and $h(\mu^k\lambda) = d_4 = x$ for $k \geq 4$. Every word from the set $\{\lambda, \mu\}^*\lambda$ can be uniquely factorized by words $\mu^k\lambda$, $k \geq 0$, whence the mapping $h$ is totally defined. We also consider the morphism $\overline{h}\colon \Delta^* \to \{\lambda, \mu\}^*\lambda$ defined by the rule $\overline{h}(d_k) = \mu^{k-1}\lambda$.

Now we take the constructed above DFA $\mathscr{B} = \langle P, \Delta, \tau\rangle$ with the state set $P = \{p_1, p_2, s'\}$. We build $\mathscr{D} = \langle D, \{\mu, \lambda\}, \tau_2\rangle$ with a unique sink state $\zeta_2$.

We associate each state $p_i$ of the automaton $\mathscr{B}$ with a 4-element set of states $P_i = \{p_{i,1}, \ldots, p_{i,4}\}$ of the automaton $\mathscr{D}$. Namely, the states $p_{i,2}, p_{i,3}, p_{i,4}$ are copies of the state $p_i$ associated with $p_{i,1}$. The action of the letter $\mu$ is defined in the following way: $\tau_2(p_{i,k}, \mu) = p_{i,k+1}$ for $k \leq 3$, and $\tau_2(p_{i,4}, \mu) = p_{i,4}$. We put $D = P_1 \cup P_2 \cup \{\zeta_2\}$, where $\zeta_2$ is a unique sink state, $P_1 = \{p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4}\}$, $P_2 = \{p_{2,1}, p_{2,2}, p_{2,3}, p_{2,4}\}$. The action of the letter $\lambda$ is defined by the rules:

- if $\tau(p_i, d_k) = s'$, then $\tau_2(p_{i,k}, \lambda) = \zeta_2$;
- if $\tau(p_i, d_k) = p_j$, then $\tau_2(p_{i,k}, \lambda) = p_{j,1}$.

The latter rule means that if there is a transition from $p_i$ to $p_j$ labeled by the letter $d_k$, then there is a transition from $p_{i,1}$ to $p_{j,1}$ labeled by the word $\mu^{k-1}\lambda$.

We build analogously $\mathscr{C} = \langle C, \{\mu, \lambda\}, \tau_1\rangle$ with a unique sink state $\zeta_1$. Every state $f_t \in F_i$ from $\mathscr{A}$ is associated with the set $P_{f_t} = \{f_{t,1}, \ldots, f_{t,4}\}$ of states of $\mathscr{C}$. We associate states $q_i$, $h$ and every $g_j \in G_i$ of $\mathscr{A}$ with 4-element sets $P_{q_i} = \{q_{i,1}, \ldots, q_{i,4}\}$, $H = \{h_1, h_2, h_3, h_4\}$ and $P_{g_j} = \{g_{j,1}, \ldots, g_{j,4}\}$ of states of $\mathscr{C}$ respectively. Finally, we put $C = \bigcup_{f_t, g_j}(P_{f_t} \cup P_{g_j}) \cup P_{q_i} \cup H \cup \{\zeta_1\}$, where $\zeta_1$ is a unique sink state. The action of the transition function $\tau_1$ is defined similarly to the definition of $\tau_2$.

Figure 3 illustrates the automata $\mathscr{D}$ (left) and $\mathscr{C}$ (right). The actions of $\mu$ and $\lambda$ are shown by solid and dashed arrows, respectively. For compactness, we do not

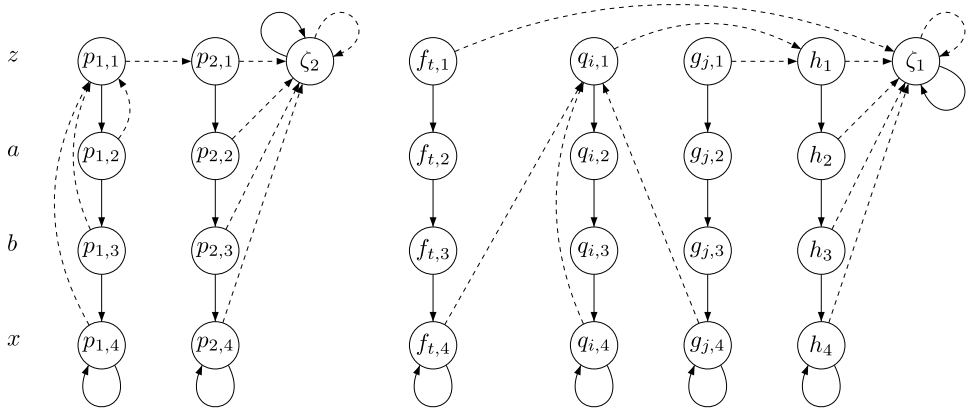


Fig. 3. The automata $\mathscr{D}$ (left) and $\mathscr{C}$ (right).

show some transitions labeled by $\lambda$ in $\mathscr{C}$. Nevertheless, the action of $\lambda$ is defined on each state in $\mathscr{C}$. The resulting automata $\mathscr{C}$ and $\mathscr{D}$ have $4(|Q|-1)+1$ and 9 states respectively, where $|Q|$ is the cardinality of the state set of $\mathscr{A}$.

**Lemma 5.** $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$ *if and only if* $\mathrm{Syn}(\mathscr{C}) = \mathrm{Syn}(\mathscr{D})$.

**Proof.** It is convenient to organize the DFA $\mathscr{D}$ as a table. The set $P_i$ is called the *i-th column* of the set $D$. For each $k = 1, \ldots, 4$, the set $R_k = \{p_{1,k}, p_{2,k}\}$ is called the *k-th row* of the set $D$. The DFA $\mathscr{C}$ will be viewed as a table as well. Sets $P_{f_t}$, $P_{g_j}$, $P_{q_i}$, $H$ are columns. One can easily write the $k$-th row for each $k = 1, \ldots, 4$.

Assume that $\mathrm{Syn}(\mathscr{A}) \neq \mathrm{Syn}(\mathscr{B})$. From the proof of Lemma 2 it follows that the word $w = xw'z$ with $w' \in \bigcap_i L[M_i]$ is reset for $\mathscr{A}$ and it is not reset for $\mathscr{B}$. Thus, $\overline{h}(w) \in \mathrm{Syn}(\mathscr{C})$ and $\overline{h}(w) \notin \mathrm{Syn}(\mathscr{D})$ since $\tau_2(p_{1,1}, \overline{h}(w)) = p_{2,1} \neq \zeta_2$. So $\mathrm{Syn}(\mathscr{C}) \neq \mathrm{Syn}(\mathscr{D})$.

Assume now that $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$. We show that every minimal reset word of $\mathscr{C}$ is reset for $\mathscr{D}$ and every minimal reset word of $\mathscr{D}$ is reset for $\mathscr{C}$. Let $u$ be a minimal reset word of $\mathscr{C}$. Every word $u \in \{\mu\}^*$ is not in $\mathrm{Syn}(\mathscr{C})$ since $\mu$ brings each column to its subset. Thus, $u$ contains some factor from $\{\lambda\}^+$. The automaton $\mathscr{C}$ possesses a unique sink state $\zeta_1$. Hence $\mathscr{C}$ is synchronized to $\zeta_1$. Furthermore, all transitions leading to $\zeta_1$ are labeled by $\lambda$, and $\zeta_1$ is fixed by $\mu$ and $\lambda$. Thus if $u$ does not end up with $\lambda$, then it is not a minimal reset word. We have $u \in \{\lambda, \mu\}^*\lambda$, i.e., $u = v\lambda$ for some $v \in \{\lambda, \mu\}^*$. Let us note that $\lambda^2$ appears in $u$ as a factor (otherwise, $u \notin \mathrm{Syn}(\mathscr{C})$). Indeed, if $u = \mu^{k_1}\lambda\mu^{k_2}\lambda \ldots \mu^{k_s}\lambda$ then $u$ maps the state set of $\mathscr{C}$ to a subset of the 1-st row containing a state different from $\zeta_1$, so $u \notin \mathrm{Syn}(\mathscr{C})$. Let us assume that the last letter of $v$ is $\lambda$ and $\lambda^2$ is not a factor of $v$, so $u = \mu^{k_1}\lambda\mu^{k_2}\lambda \ldots \mu^{k_s}\lambda^2$ where $k_i \neq 0$ for $i > 1$. Since $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B})$ and $\bigcap_i L[M_i] = \emptyset$, thus the image of the state set of $\mathscr{C}$ under the action of $\mu^{k_1}\lambda\mu^{k_2}\lambda \ldots \mu^{k_s}\lambda$ contains $q_{r,1}$ or $g_{r,1}$ for some $r$ (since $\mu^{k_1}\lambda$ maps $q_{i,4}$ to $q_{i,1}$ identifying an initial state in $\mathscr{A}$). So $u \notin \mathrm{Syn}(\mathscr{C})$ since $\lambda$ does not map $q_{r,1}$ or $g_{r,1}$ to $\zeta_1$. Therefore, $\lambda^2$ is a factor of $v$,

then by the definition of the transition functions of $\mathscr{C}$ and $\mathscr{D}$ we have $u \in \mathrm{Syn}(\mathscr{C})$ and $u \in \mathrm{Syn}(\mathscr{D})$. So the inclusion $\mathrm{Syn}(\mathscr{C}) \subseteq \mathrm{Syn}(\mathscr{D})$ takes place. The opposite inclusion $\mathrm{Syn}(\mathscr{D}) \subseteq \mathrm{Syn}(\mathscr{C})$ is verified analogously. □

Lemma 5 implies **PSPACE**-completeness of the problem SYN-EQUALITY restricted to a binary alphabet.

**Theorem 6 (Theorem 4, [9]).** *SYN-EQUALITY restricted to a binary alphabet case is* **PSPACE***-complete.*

As a corollary, we immediately obtain the following statement.

**Proposition 7 (Proposition 1, [9]).** *Given an integer $\ell > 0$ and a synchronizing DFA $\mathscr{A}$. The problem of checking the inequality $rc(\mathrm{Syn}(\mathscr{A})) \leq \ell$ is in* **PSPACE***.*

So we reduced FINITE AUTOMATA INTERSECTION to the complement of SYN-EQUALITY restricted to a binary case alphabet as follows. For an arbitrary instance of FINITE AUTOMATA INTERSECTION one may build the corresponding automata $\mathscr{C}$ and $\mathscr{D}$ over a binary alphabet $\{\lambda, \mu\}$ such that $\bigcap_{i=1}^{n} L[M_i] \neq \varnothing$ if and only if $\mathrm{Syn}(\mathscr{C}) = \mathrm{Syn}(\mathscr{D})$.

The set of all words synchronizing a fixed subset $H \subseteq D$ of the state set of $\mathscr{D}$ is defined as follows:

$$\mathcal{R}(H) = \{v \in \{\lambda, \mu\}^* \mid H \,.\, v = \{\zeta_2\}\}.$$

Since $\zeta_2$ is a unique sink state in $\mathscr{D}$, every reset word for $\mathscr{D}$ maps every subset of $D$ to $\{\zeta_2\}$. Let us note that

$$\mathcal{R}(\{p_{2,1}\}) = \mathcal{R}(\{p_{2,2}\}) = \mathcal{R}(\{p_{2,3}\}) = \mathcal{R}(\{p_{2,4}\});$$
$$\mathcal{R}(\{p_{1,2}\}) = \mathcal{R}(\{p_{1,3}\}) = \mathcal{R}(\{p_{1,4}\}).$$

These equalities imply that the language of reset words of $\mathscr{D}'$ (defined on Fig. 4) coincides with the language of reset words of $\mathscr{D}$. Indeed, due to the equalities above we can merge states $p_{2,1}$, $p_{2,2}$, $p_{2,3}$, $p_{2,4}$ into a unique state $p_2$ and merge states $p_{1,2}$, $p_{1,3}$, $p_{1,4}$ into a unique state $p_0$. Solid arrows still denote the action of $\mu$ while dashed arrows stand for $\lambda$. In what follows we consider the DFA $\mathscr{D}'$ instead of $\mathscr{D}$.

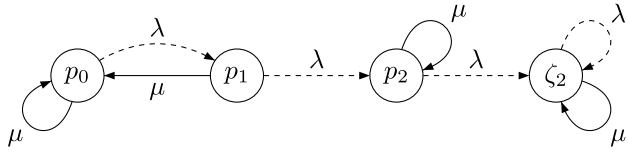


Fig. 4. The automaton $\mathscr{D}'$.

**Lemma 8.** *Let $J = \mathrm{Syn}(\mathscr{C})$. The equality $\bigcap_{i=1}^{n} L[M_i] = \varnothing$ takes place if and only if $rc(J) \leq 4$.*

**Proof.** Let $\mathscr{E} = \langle P, \{\lambda, \mu\}, \gamma\rangle$ be an MSA for $J$. Assume that $\bigcap_{i=1}^{n} L[M_i] = \varnothing$. By Lemma 2 we get that $\mathrm{Syn}(\mathscr{A}) = \mathrm{Syn}(\mathscr{B}) = I$ with $I = (\Sigma \cup \{x\})^* z \Delta^+$. By Lemma 5 we have $J = \mathrm{Syn}(\mathscr{C}) = \mathrm{Syn}(\mathscr{D}')$. Let us note that $\lambda^3 \in \mathrm{Syn}(\mathscr{D}')$. Furthermore, $\lambda, \lambda^2 \notin \mathrm{Syn}(\mathscr{D}')$. It means that $P.\lambda^3 \subsetneq P.\lambda^2 \subsetneq P.\lambda \subsetneq P$. Hence $|P| \geq 4$. Therefore, $rc(J) \geq 4$. On the other hand, $\mathrm{Syn}(\mathscr{D}') = J$. Thus, $\mathscr{D}'$ is an MSA for $J = \mathrm{Syn}(\mathscr{C})$, i.e., $rc(J) = 4$.

Let us assume now that $\bigcap_{i=1}^{n} L[M_i] \neq \varnothing$. We are going to show that $rc(J) > 4$ in this case. Let $u$ be a minimal reset word for $\mathscr{C}$. By the arguments from the proof of Lemma 5 and by the construction of $\mathscr{C}$, one may note that $u$ can be factorized as $u = v\lambda$ for some $v \in \{\mu, \lambda\}^+$. Also $\lambda^2$ is necessarily a factor of $u$. Indeed, let us assume that $u = \mu^{k_1}\lambda\mu^{k_2}\lambda \ldots \mu^{k_s}\lambda$ with $k_i \neq 0$ for $i > 1$. If the image of the state set of $\mathscr{C}$ under the action of $u$ contains $q_{r,1}$ or $g_{r,1}$ for some $r$, then $u \notin \mathrm{Syn}(\mathscr{C})$. If $u$ maps the state set of $\mathscr{C}$ to a subset of $\bigcup_t \{f_{t,1}\} \cup \{h_1, \zeta_1\}$, then we have to apply $\lambda$ to map this subset to $\zeta_1$. All in all we have $u \notin \mathrm{Syn}(\mathscr{C})$. Moreover, by the definition of the transition function of $\mathscr{C}$ we have $\lambda^3 \in J$ while $\lambda, \lambda^2 \notin J$. From the arguments above it follows that every MSA for $J$ has at least 4 states. Arguing by contradiction, let us assume that there exists a 4-state automaton $\mathscr{E} = \langle P, \{\lambda, \mu\}, \gamma\rangle$ with $\mathrm{Syn}(\mathscr{E}) = J$. Without loss of generality suppose that $P = \{0, 1, 2, 3\}$. We define the action of $\lambda$ on the state set $P$. Since $\lambda^3 \in \mathrm{Syn}(\mathscr{E})$ and $\lambda, \lambda^2 \notin \mathrm{Syn}(\mathscr{E})$, there is a unique up to isomorphism way to define the action of $\lambda$ on the states from the state set $P$ (see Fig. 5).

The word $\lambda^2\mu\lambda$ is in $\mathrm{Syn}(\mathscr{C})$, i.e., $\lambda^2\mu\lambda \in J$. By the definition of the action of $\lambda$ in $\mathscr{E}$ we get that $\gamma(P, \lambda^2) = \{2, 3\}$. On the other hand, $\lambda^2\mu \notin \mathrm{Syn}(\mathscr{C})$, thus $\lambda^2\mu \notin \mathrm{Syn}(\mathscr{E})$. Hence $\{2, 3\}$ under the action of $\mu$ is mapped to a two-element subset which is translated by $\lambda$ into $\{3\}$. So we need to guarantee the equality $\gamma(\{2, 3\}, \mu) = \{2, 3\}$. The action of $\mu$ at states 2 and 3 can be defined in two possible ways such that the last equality is true (see Fig. 6).

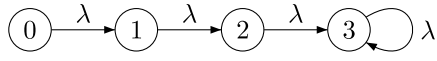


Fig. 5. The action of $\lambda$ in $\mathscr{E}$.

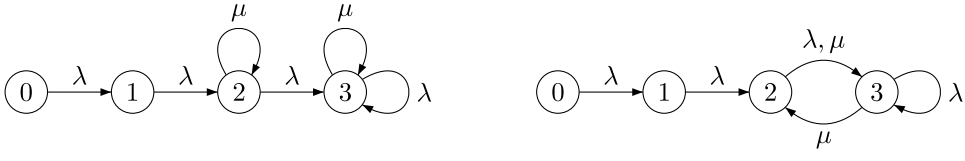


Fig. 6. The action of $\mu$ at states 2 and 3 in the DFA $\mathscr{E}$.

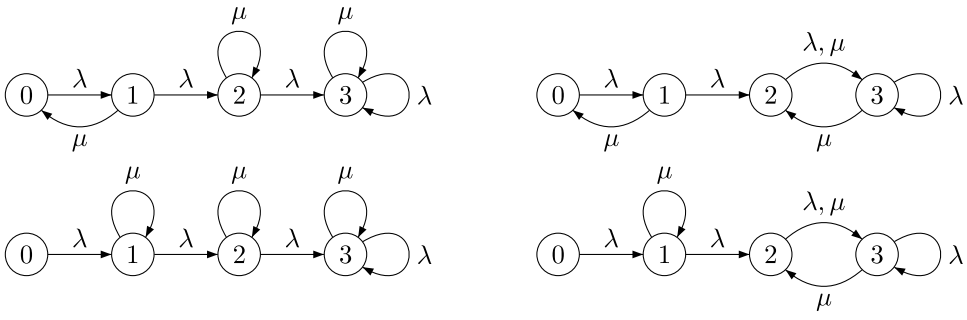


Fig. 7. Possible ways to define the action of $\mu$ at 1 in $\mathscr{E}$.

Note that $\lambda\mu\lambda \notin \mathrm{Syn}(\mathscr{C})$ since $\tau_1(q_{i,4}, \lambda\mu\lambda) = \tau_1(q_{i,2}, \lambda) \neq \zeta_1$. Therefore, $\lambda\mu\lambda$ is not a reset word for $\mathscr{E}$. Since we have $\gamma(\{1,2,3\}, \lambda\mu\lambda) = \{3\}$ for each variant from Fig. 6, the word $\lambda\mu\lambda$ should map the state 0 to a state different from 3. However $\gamma(0,\lambda) = 1$, thus 1 under the action of $\mu$ is mapped to either 0 or 1. All in all, for each variant from Fig. 6, we get two ways to define the image of 1 under the action of $\mu$ (see Fig. 7).

It remains to define the image of 0 under the action of $\mu$. Let us note that $\mu^2\lambda^2 \notin \mathrm{Syn}(\mathscr{C})$ since $\tau_1(q_{i,2}, \mu^2\lambda^2) = \tau_1(q_{i,4}, \lambda^2) = h_1 \neq \zeta_1$. Since $\gamma(\{2,3\}, \mu^2\lambda^2) = \{3\}$, one of the equalities, $\gamma(0,\mu) = 0$ or $\gamma(0,\mu) = 1$, is required to take place. Furthermore, we have $\gamma(\{1,2,3\}, \mu^2\lambda^2) = \{3\}$ in the third and forth variants from Fig. 7. It means that there exists only one possibility to put $\gamma(0,\mu) = 0$ for these variants. So we have six automata over the alphabet $\{\lambda, \mu\}$ shown in Fig. 8. It remains to check whether $J$ could coincide with the set of reset words for one of these DFAs.

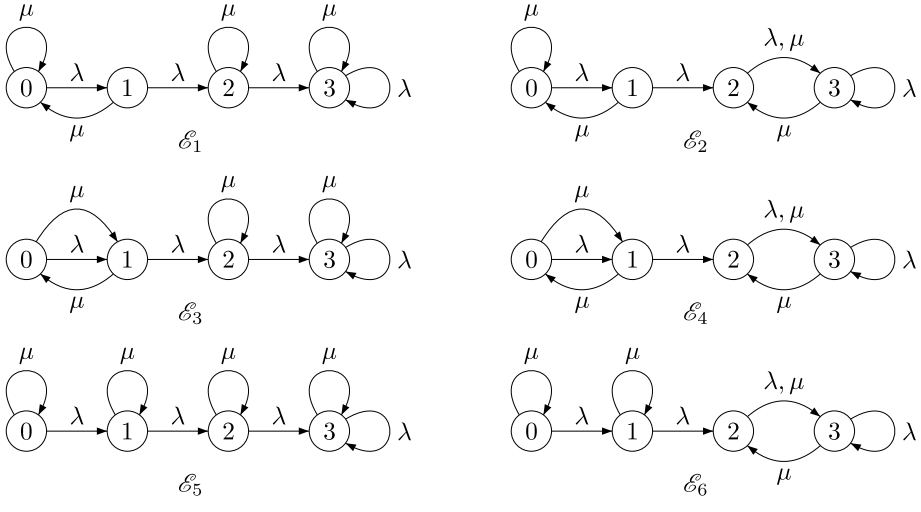


Fig. 8. Possible candidates for $\mathscr{E}$.

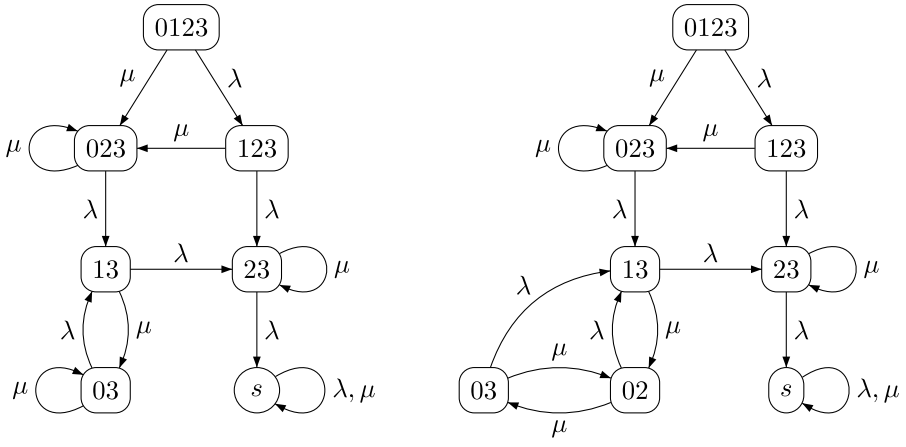


Fig. 9. The power automata $\mathcal{P}(\mathscr{E}_1)$ and $\mathcal{P}(\mathscr{E}_2)$.

The DFA $\mathscr{E}_1$ from Fig. 8 is isomorphic to $\mathscr{D}'$ for which we have $\mathrm{Syn}(\mathscr{D})' = \mathrm{Syn}(\mathscr{D})$. So $\mathrm{Syn}(\mathscr{E}_1) = \mathrm{Syn}(\mathscr{D})$. By the assumption $\bigcap_{i=1}^{n} L[M_i] \neq \varnothing$, hence by Lemmas 2 and 5 we have $J = \mathrm{Syn}(\mathscr{C}) \neq \mathrm{Syn}(\mathscr{D})$. Therefore, $\mathscr{E}_1$ is not an MSA for $J$. For $\mathscr{E}_1$ and $\mathscr{E}_2$ the equality $\mathrm{Syn}(\mathscr{E}_1) = \mathrm{Syn}(\mathscr{E}_2)$ takes place. It can be easily checked by the construction of power automata $\mathcal{P}(\mathscr{E}_1)$ and $\mathcal{P}(\mathscr{E}_2)$ (see Fig. 9). It implies that $\mathscr{E}_2$ cannot be an MSA for $J$.

By the assumption $c \notin \bigcap_{i=1}^{n} L[M_i]$ for all $c \in \Sigma$. It means that $b \notin L[M_j]$ or, equivalently, $\delta_j(q_j, b) \in Q_j \backslash F_j$ for some index $j$. Take the word $xbz$. By the definition of the transition function $\varphi$ of $\mathscr{A}$ we have $\varphi(Q, xbz) = \{h, s\}$, so $xbz \notin \mathrm{Syn}(\mathscr{A})$. By the definition of the morphism $\overline{h}\colon \Delta^* \to \{\lambda, \mu\}^*\lambda$ we have $\overline{h}(xbz) = \mu^3\lambda\mu^2\lambda\lambda$. By the definition of the transition function $\varphi_2$ of $\mathscr{C}$ we get $\varphi_2(q_{j,1}, \mu^3\lambda\mu^2\lambda) = \varphi_2(q_{j,1}, \mu^2\lambda) = p_{t,1}$, where $p_t = \delta_j(q_j, b)$ and $p_t \notin F_j$. Hence $\varphi_2(q_{j,1}, \mu^3\lambda\mu^2\lambda\lambda) = \varphi_2(p_{t,1}, \lambda) = h_1$. So we obtain that $\mu^3\lambda\mu^2\lambda\lambda \notin \mathrm{Syn}(\mathscr{C})$. Therefore, $\mu^3\lambda\mu^2\lambda\lambda \notin J$, but it is easy to see that $\mu^3\lambda\mu^2\lambda\lambda \in \mathrm{Syn}(\mathscr{E}_3)$. Hence $\mathscr{E}_3$ can not be an MSA for $J$. Analogously, $\mathscr{E}_4$ can not be an MSA for $J$ as well.

Note that $(\mu\lambda)^3 \in \mathrm{Syn}(\mathscr{E}_5)$ and $(\mu\lambda)^3 \in \mathrm{Syn}(\mathscr{E}_6)$. By the definition of the morphism $h\colon \{\lambda, \mu\}^*\lambda \to \Delta^*$, we have $h((\mu\lambda)^3) = (h(\mu\lambda))^3 = a^3$. But the word $a^3$ is not reset for $\mathscr{A}$ (see Fig. 1). By the definition of the transition function of $\mathscr{C}$ it implies that $\overline{h}(a^3) \notin \mathrm{Syn}(\mathscr{C})$, that is $(\mu\lambda)^3 \notin J$. In this way neither $\mathscr{E}_5$, nor $\mathscr{E}_6$ can be an MSA for $J$. We have considered all possible candidates for a 4-state MSA for $J$. Each automaton $\mathscr{E}_i$ cannot be chosen as an MSA for $J$. Also it is known that $rc(J) \geq 4$. Finally, we have $rc(J) > 4$. □

Now we are in position to state the main result of the paper. Lemma 8 and Proposition 7 imply the following theorem.

**Theorem 9.** *RESET-INEQUALITY restricted to a binary alphabet is* **PSPACE***-complete for* $\ell = 4$.

## 5. State Complexity of Languages Related to Slowly Synchronizing Automata

In this section, we study series of "slowly" synchronizing automata. The first construction belongs to Černý [3], the others are taken from [1]. Černý constructed for each $n > 1$ a synchronizing automaton automaton $\mathscr{C}_n$ with $n$ states, two input letters and reset threshold $(n-1)^2$. Recall the definition of $\mathscr{C}_n$. If we denote the states of $\mathscr{C}_n$ by $0, 1, \ldots, n-1$ and the input letters by $a$ and $b$, the actions of these letters are defined as follows:
$i \,.\, b = i + 1$ for $0 \leq i \leq n-2$, and $(n-1) \,.\, b = 0$;
$i \,.\, a = i$ for $0 \leq i \leq n-2$, and $(n-1) \,.\, a = 0$.

Due to [8] we have the following complexity result.

**Proposition 10 (Proposition 2, [8]).** $sc(Syn(\mathscr{C}_n)) = 2^n - n$.

In particular, it has been shown that each non-empty subset $H \subseteq Q$ is *reachable* in $\mathscr{C}_n$, that is $Q \,.\, w = H$ for some $w \in \Sigma^*$. It means that $\mathscr{C}_n$ is *completely reachable*. Recently Don [4, Theorem 1] found a sufficient condition for complete reachability that applies to the automata $\mathscr{C}_n$. Another sufficient condition was presented in Ref. [2] that both simplified and generalized Don's one.

Two other series of slowly synchronizing automata has been considered in Ref. [8]. Let us recall the corresponding constructions of $n$-state automata $\mathscr{L}_n$ and $\mathscr{V}_n$ (see Fig. 10 and Fig. 11).

**Proposition 11 (Proposition 3, [8]).** $sc(Syn(\mathscr{L}_n)) = 2^n - n$.

**Proposition 12 (Proposition 4, [8]).** $sc(Syn(\mathscr{V}_n)) = 2^n - n$.

The automata $\mathscr{L}_n$ and $\mathscr{V}_n$ are completely reachable as well. Now we check other series of slowly synchronizing automata for complete reachability and calculate for

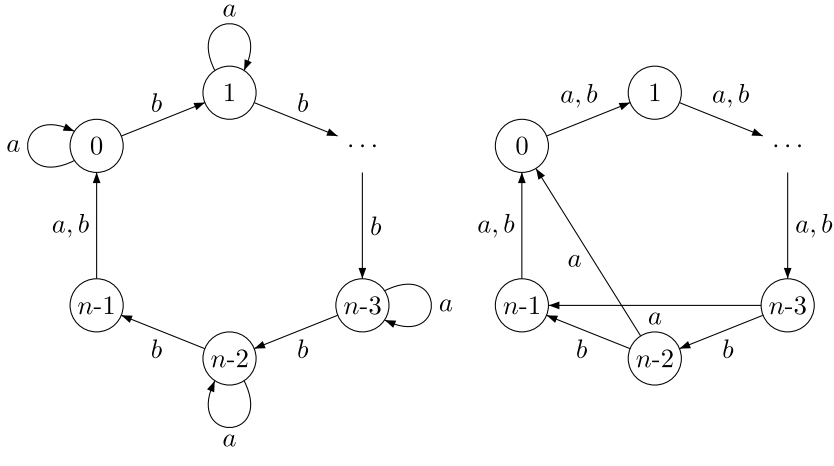


Fig. 10. The automata $\mathscr{C}_n$ (left) and $\mathscr{L}_n$ (right).

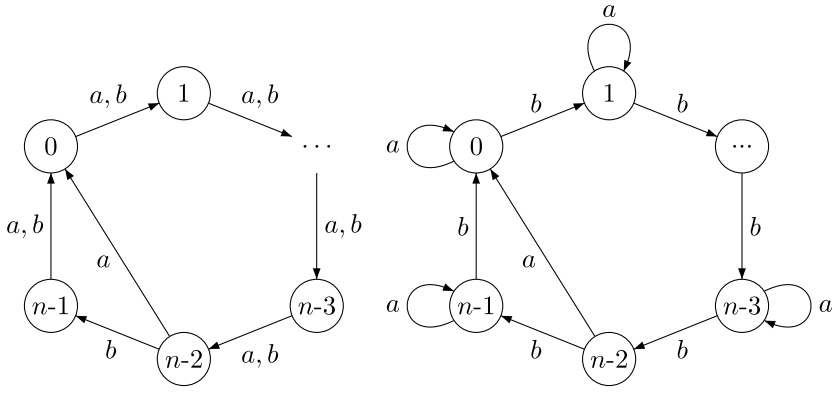


Fig. 11. The automata $\mathscr{Y}_n$ (left) and $F_n$ (right).

several of them the state complexity and the reset complexity of its languages of reset words. Recall the definition of the automaton $\mathscr{F}_n$ from [1]. If we denote the states of $\mathscr{F}_n$ with $n = 2m + 1$ by $0, 1, \ldots, n-1$, where $n$ is odd, and the input letters by $a$ and $b$, the actions of these letters are defined as follows:
$i \,.\, b = i + 1$ for $0 \le i \le n-2$, and $(n-1) \,.\, b = 0$;
$i \,.\, a = i$ for $0 \le i \le n-3$ or $i = n-1$, and $(n-2) \,.\, a = 0$.
The automaton $\mathscr{F}_n$ is presented on the Fig. 11.

We define the functions $d(p,q) : Q \times Q \to \mathbb{R}$ and $qd(p,q) : Q \times Q \to \mathbb{R}$ of *distance* and *quasi-distance* between states $p$ and $q$ as follows (without loss of generality assume that $p < q$):

$$d(p,q) = \min\{q-p, n+p-q\}, \tag{1}$$

$$qd(p,q) = \begin{cases} d(p,q), & \text{if } d(p,q) \text{ is even,} \\ n - d(p,q), & \text{if } d(p,q) \text{ is odd.} \end{cases} \tag{2}$$

For example, $d(1, n-1) = qd(1, n-1) = 2$, $d(0, n-1) = 1$, $qd(0, n-1) = n-1$. If $d(p', q')$ is odd for some pair $\{p', q'\}$, then $qd(p', q') > qd(p, q)$ for any $\{p, q\}$ with even distance. Given a two-state subset $T = \{p, q\}$, $p < q$. Let $d = d(p,q)$, $t = qd(p,q)$, and define the parameter $\alpha$ by the following rules:

- if $d$ is even, then $\alpha = \begin{cases} (n - q + n - 2) \mod n, \text{ for } d = n - q + p, \\ 0, \text{ for } T = \{d-2, n-2\}, \\ n - 2 - p, \text{ otherwise} \end{cases}$
- if $d$ is odd, then $\alpha = \begin{cases} (n - q + n - 2) \mod n, \text{ for } d = q - p, \\ n - 2 - p, \text{ for } d = n - q + p. \end{cases}$

The main goal of introducing such a parameter $\alpha$ is to decrease the quasi-distance between $p$ and $q$ by 2 after applying the word $b^\alpha a$ to $T$. Indeed, $T \,.\, b^\alpha = \{n-2, d-2\}$

for even $d$, and $T\,.\,b^\alpha = \{n-d-2, n-2\}$ for odd $d$. Hence $T\,.\,b^\alpha a = \{n-2, t-2\}\,.\,a = \{0, t-2\}$. Note that, the word

$$w = b^\alpha a (b^{n-2}a)^{t/2-1} \tag{3}$$

synchronizes $T$. Indeed, $T\,.\,b^\alpha a = \{0, t-2\}$. The state 0 is fixed by the word $u = b^{n-2}a$, and the state $t-2$ under the action of the word $u$ moves to the state $t-4$. Thus, after applying the word $u^{t/2-1}$ to the set $\{0, t-2\}$ we obtain the set $\{0\}$. Hence $T\,.\,w = \{0\}$.

The function $qd(H)$ for a subset $H \subseteq Q$ is defined by

$$qd(H) = \min_{p,q\in H, p\neq q} qd(p,q). \tag{4}$$

Given two subsets $H$ and $S$ of $Q$ of size at least two, let us find $qd = \min\{qd(H), qd(S)\}$. Denote by $\{p,q\}$ the pair with the quasi-distance $qd$ (if there are several such pairs, we construct for them corresponding words by (3) and choose the pair with the shortest word $w$). Without loss of generality we may assume $\{p,q\} \subseteq H$. The following auxiliary claim will be of use.

**Lemma 13.** *Let $\{p,q\}$ be the pair chosen as above, and $w$ be the word constructed for the pair $\{p,q\}$ by* (3)*. No other pair in $H$ or $S$ is synchronized by $w$.*

**Proof.** Arguing by contradiction, suppose there is a pair $T' = \{p', q'\}$ $(p' < q')$ either in $H$ or in $S$ such that $p'\,.\,w = q'\,.\,w$ and either $p'$ is different from $p$ or $q'$ is different from $q$. Let $t = qd(p,q)$, $t' = qd(p',q')$. By the definition of $t$ we have $t' \geq t$. Suppose $0 \notin T'\,.\,b^\alpha a = \{p'', q''\}$. We get $p''\,.\,b^{n-2}a = (p''-2+n) \mod n$, and $q''\,.\,b^{n-2}a = (q''-2+n) \mod n$. The latter equalities mean that the distance between the states $p''$ and $q''$ does not change, thus, for the word $w$ to synchronize $T'$, it is necessary that $p'' = q''$. Since $b$ is a permutation letter, the only possibility for this to happen is $p'\,.\,b^\alpha a = q'\,.\,b^\alpha a = 0$, a contradiction with the supposition $0 \notin \{p'', q''\}$. So, $0 \in T'\,.\,b^\alpha a$. If $n-2 \in T'\,.\,b^\alpha$, then $T'\,.\,b^\alpha = \{n-2, t'-2\}$. If $t' > t$, then $T'\,.\,w = \{0, \frac{t'-t}{2}\}$, a contradiction. Thus $t' = t$, but in this case $T'\,.\,b^\alpha = T\,.\,b^\alpha$. Since $b$ is a permutation letter, we obtain $T = T'$. Since $p < q$ and $p' < q'$ we have $p = p'$, $q = q'$. A contradiction. So we have $0 \in T'\,.\,b^\alpha$. Hence $T'\,.\,b^\alpha = \{0, t'-2\}$. However $T'\,.\,w = \{0, t'-2-(\frac{t}{2}-1)\cdot 2\} = \{0, t'-t+2\}$, and $t'-t+2 > 0$ even in the case $t' = t$. Again a contradiction. □

Now we are in position to prove the main result of this section.

**Proposition 14.** *$sc(Syn(\mathscr{F}_n)) = 2^n - n$ for every odd $n > 3$.*

**Proof.** Construct for the automaton $\mathscr{F}_n = \langle Q, \Sigma, \delta\rangle$ its power automaton $\mathcal{P}(\mathscr{F}_n)$. Let $H \subseteq Q$ be a given subset, denote $\delta^{-1}(H,u) = \{q \in Q \mid \delta(q,u) \in H\}$ for $u \in \Sigma^*$.

First we check that all nonempty subsets $H \subseteq Q$ are reachable. By induction on $k = |H|$. Case $|H| = n$ is clear: the state set $Q$ of the automaton $\mathscr{F}_n$ is the initial state of $\mathcal{P}$. Assume that any subset with cardinality $1 < k \leq n$ is reachable. Now we

verify that all subsets $H$ with $|H| = k-1$ are reachable. Let $H = \{p_1, p_2, \ldots, p_{k-1}\}$, and $p_i < p_{i+1}$ for all $1 \leq i \leq k-2$.

One may find a nonnegative integer $\ell$ such that $\delta^{-1}(H, b^\ell) = H'$ and $0 \in H'$. Since $n$ is odd, there is an integer $m$ such that $H'' = \delta^{-1}(H', b^m)$, $n-2 \notin H''$ and $0 \in H''$. Since $b$ is a permutation letter, we have $|H''| = |H'| = k-1$. The subset $H'' \cup \{n-2\}$ contains $k$ states, therefore, this subset is reachable from $Q$ by induction hypothesis: $Q \,.\, w = H'' \cup \{n-2\}$ for some $w \in \Sigma^*$. Clearly $H'' \,.\, a = H''$, $(n-2) \,.\, a = 0$ and $0 \in H''$ by the choice of $m$, thus $Q \,.\, wab^m b^\ell = H'' \,.\, b^m b^\ell = H$ and $H$ is reachable from $Q$. Recall that all singletons are merged into a unique sink terminal state in $\mathcal{P}(\mathscr{F}_n)$, hence the automaton $\mathcal{P}(\mathscr{F}_n)$ consists of $2^n - 1 - (n-1) = 2^n - n$ reachable states. Thus, the minimal automaton recognizing $Syn(\mathscr{F}_n)$ has at most $2^n - n$ states. Furthermore, $\mathscr{F}_n$ is completely reachable.

Next we prove that any two states of $\mathcal{P}(\mathscr{F}_n)$ different from the terminal one are inequivalent. Take two arbitrary subsets $H$ and $S$ of $Q$ such that $H \neq S$. We verify that there exists a word $w \in \Sigma^*$ such that $H \,.\, w \neq S \,.\, w$ and at least one of the equalities $|H \,.\, w| = |H| - 1$ and $|S \,.\, w| = |S| - 1$ holds. Let us find $t = \min\{qd(H), qd(S)\}$. Denote by $\{p, q\}$ the pair yielding the minimum of quasi-distance for $H$ and $S$ (the pair is chosen as above). Construct the corresponding word $w = b^\alpha a (b^{n-2}a)^{t/2-1}$ for this pair. Without loss of generality we may assume $\{p, q\} \subseteq H$.

**Case 1:** $p, q \in H \backslash S$. If $S \subsetneq H$, then $|S| \leq |H| - 2$. By Lemma 13 we have $|H \,.\, w| = |H| - 1$, and $|S \,.\, w| = |S|$. Thus

$$|S \,.\, w| = |S| \leq |H| - 2 = |H \,.\, w| - 1 < |H \,.\, w|.$$

Therefore $S \,.\, w \neq H \,.\, w$. Suppose now $S \backslash H \neq \varnothing$. We show that either $(H \backslash S) \,.\, w$ and $(S \backslash H) \,.\, w$ do not intersect or $(H \backslash S) \,.\, w \cap (S \backslash H) \,.\, w = \{0\}$. Obviously, $(H \backslash S) \,.\, b^\alpha$ and $(S \backslash H) \,.\, b^\alpha$ do not intersect. By the definition of $\alpha$ we have $\{p, q\} \,.\, b^\alpha = \{n-2, t-2\}$, and $\{p, q\} \,.\, b^\alpha a = \{0, t-2\}$. Since $n-2 \in (H \backslash S) \,.\, b^\alpha$ we have $n-2 \notin (S \backslash H) \,.\, b^\alpha$. Hence, $(S \backslash H) \,.\, b^\alpha a = (S \backslash H) \,.\, b^\alpha$. Thus, the subsets $(H \backslash S) \,.\, b^\alpha a$ and $(S \backslash H) \,.\, b^\alpha a$ can have at most one common element, namely 0. For each $r \in Q$ such that $r \neq 0$, we have $r \,.\, b^{n-2}a = (n + r - 2) \mod n$, and for $r = 0$ we have $r \,.\, b^{n-2}a = 0$. Thus the word $b^{n-2}a$ shifts by 2 all the states different from 0. Moreover, through $t/2 - 1$ steps no state different from $p$ and $q$ moves to 0, otherwise we would get another pair synchronized by $w$, which contradicts Lemma 13. The latter argument implies that the subsets $(H \backslash S) \,.\, w$ and $(S \backslash H) \,.\, w$ can have at most one common element 0. If $(H \backslash S) \,.\, w \neq \{0\}$ or $(S \backslash H) \,.\, w \neq \{0\}$, then obviously $H \,.\, w \neq S \,.\, w$. It remains to study the case when $(H \backslash S) \,.\, w = (S \backslash H) \,.\, w = \{0\}$. By Lemma 13 it is possible if and only if $H \backslash S = \{p, q\}$ and $S \backslash H = \{r\}$ for some $r \in Q$. Then we have $r \,.\, w = 0$. By the definition of $w$ we get $r \,.\, b^\alpha a (b^{n-2}a)^k = 0$ for some $0 \leq k \leq t/2 - 1$. If $t = 2$, then $p \,.\, b^\alpha a = q \,.\, b^\alpha a = r \,.\, b^\alpha a = 0$ and $r \in \{p, q\}$, a contradiction. So $t > 2$. Consider subsets $\overline{H} = H \,.\, b^\alpha a (b^{n-2}a)^{t/2-2}$ and $\overline{S} = S \,.\, b^\alpha a (b^{n-2}a)^{t/2-2}$. Note that $\overline{S} \subseteq \overline{H}$ and $\overline{H} \backslash \overline{S} = \{2\}$. It remains to check that there exists a word $v \in \Sigma^*$ such that $\overline{H} \,.\, v \neq \overline{S} \,.\, v$ and at least one of the equalities

$|\overline{H} \,.\, v| = |\overline{H}| - 1$ and $|\overline{S} \,.\, v| = |\overline{S}| - 1$ holds. This situation will be studied later inside the Case 3.

**Case 2:** $p, q \in G = H \cap S$. If $S \subsetneq H$, then $|S| < |H|$. By Lemma 13 $|S \,.\, w| = |S| - 1$ and $|H \,.\, w| = |H| - 1$, so $S \,.\, w \neq H \,.\, w$. The case $H \subsetneq S$ is considered symmetrically. So we may assume $S \backslash H \neq \varnothing$ and $H \backslash S \neq \varnothing$. We show that $(H \backslash S) \,.\, w \cap (S \backslash H) \,.\, w = \varnothing$. Apply $w$ to $H \backslash S$ and $S \backslash H$. Let us note that $(H \backslash S) \,.\, b^\alpha$ and $(S \backslash H) \,.\, b^\alpha$ have empty intersection. Next we apply the letter $a$ to $(H \backslash S) \,.\, b^\alpha$ and $(S \backslash H) \,.\, b^\alpha$. All the states in these subsets are fixed by $a$. Otherwise some state moves to 0 and we came to a contradiction with Lemma 13. Besides, by the choice of the pair $\{p, q\}$ either $p \,.\, b^\alpha a = 0$ or $q \,.\, b^\alpha a = 0$. Finally, we apply $t/2 - 1$ times the word $b^{n-2}a$. Under the action of $b^{n-2}a$ the numbers of all states in both subsets decrease by 2. Moreover, through $t/2 - 1$ steps no state from $(H \backslash S) \,.\, b^\alpha a$ or $(S \backslash H) \,.\, b^\alpha a$ moves to 0, otherwise we would again get a contradiction with Lemma 13. Thus, $H \,.\, w \neq S \,.\, w$.

**Case 3:** one of the states of the pair $\{p, q\}$ belongs to $G$, and the other to $H \backslash S$. By Lemma 13 we have $|S \,.\, w| = |S|$ and $|H \,.\, w| = |H| - 1$. Let $S \backslash H \neq \varnothing$. Lemma 13 implies $0 \notin (S \backslash H) \,.\, w$. By the same argument as in the previous cases we obtain that the sets $(H \backslash S) \,.\, w$ and $(S \backslash H) \,.\, w$ do not intersect. So $H \,.\, w \neq S \,.\, w$. If $S \subsetneq H$ and moreover $|S| < |H| - 1$, then we get $|S \,.\, w| = |S| < |H| - 1 = |H \,.\, w|$, so $S \,.\, w \neq H \,.\, w$. Finally, it remains to consider the case $H = S \cup \{r\}$. We may assume $r = 0$ (otherwise we apply the word $b^{n-r}$ to $S$ and $H$). Let $S = \{q_1, q_2, \ldots, q_\ell\}$, then $H = \{0, q_1, q_2, \ldots, q_\ell\}$. We have $p = 0$ and $q = q_i$ for some $1 \leq i \leq \ell$.

Next we find a word $u$ such that subsets $H \,.\, u$ and $S \,.\, u$ satisfy the equality $H \,.\, u \backslash S \,.\, u = \{0\}$ and contain the pair $\{\overline{p}, \overline{q}\}$ yielding the minimum of quasi-distance for $H \,.\, u$ and $S \,.\, u$. This purpose can be achieved, since $b$ labels a cycle of odd length in the DFA $\mathscr{F}_n$, by playing with words of the form $w_1 = b^{n-2}a$, $w_2 = b^m$, $w_3 = b^{n-2-q}ab^{q+2}$. Note that $w_1$ shifts each state, apart from 0, by 2 on the cycle labeled by $b$ and $0 \,.\, w_1 = 0$, $w_2$ shifts each state by $m$ on the cycle labeled by $b$, $w_3$ fixes each state, apart from $q$, that is $p \,.\, w_3 = p$, for $p \neq q$, $q \,.\, w_3 = (q + 2) \mod n$. If $\overline{p}, \overline{q} \in S \,.\, u$, then we apply the argument from Case 2 and find a word $\overline{w}$ such that $S \,.\, uw \neq H \,.\, uw$ and $|H \,.\, uw| = |H \,.\, u| - 1 = |H| - 1$. Otherwise repeat the algorithm above applied to the subsets $H \,.\, u$ and $S \,.\, u$. Through the finite number of steps we will obtain the subsets $H \,.\, \overline{u}$ and $S \,.\, \overline{u}$ such that the corresponding pair $\{\overline{p}, \overline{q}\}$ is contained in $S \,.\, \overline{u}$. And this case was studied above.

So we have that for two arbitrary subsets $H$ and $S$ there exists a word $w \in \Sigma^*$ such that $H \,.\, w \neq S \,.\, w$ and at least one of the equalities $|H \,.\, w| = |H| - 1$ and $|S \,.\, w| = |S| - 1$ holds. Next we consider subsets $H \,.\, w$, $S \,.\, w$. If none of them is $\{0\}$, apply described algorithm again. It is clear that through the finite number of steps we will find a word $\overline{w}$ with the property $\overline{w} \in Syn(H) \backslash Syn(S)$ (or $\overline{w} \in Syn(S) \backslash Syn(H)$). □

There is another example of $n$-automaton among known constructions of slowly synchronizing automata that seems to be completely reachable. Namely, the

definition of the automaton $\mathscr{K}_n$ with odd $n$ provides the desired series. If we denote the states of $\mathscr{K}_n$ with $n = 2m+1$ by $0, 1, \ldots, n-1$, where $n$ is odd, and the input letters by $a$ and $b$, the actions of these letters are defined as follows:

$i \,.\, b = i+1$ for $0 \le i \le n-2$, and $(n-1) \,.\, b = 0$;
$i \,.\, a = i+1$ for $0 \le i \le n-4$ or $i = n-2$, $(n-3) \,.\, a = 0$, $(n-1) \,.\, a = 2$.

**Conjecture.** $sc(Syn(\mathscr{K}_n)) = 2^n - n$ for every odd $n > 5$.

The conjecture has been confirmed by computer experiments for $n \le 21$. In general it seems that the proof of the conjecture can be obtained using the same standard technique as has been implemented in Proposition 14. So we omit more details here because of space constraints. It has been shown in [7, Theorem 1] that the reset complexity of the languages of reset words for $\mathscr{C}_n$, $\mathscr{L}_n$ and $\mathscr{V}_n$ is equal to $n$. It means that these $n$-state automata are MSA's for the corresponding ideal languages. We prove that the reset complexity of $\mathrm{Syn}(\mathscr{F}_n)$ and $\mathrm{Syn}(\mathscr{K}_n)$ is equal to $n$ as well.

**Theorem 15.** *For every odd $n > 3$, $rc(Syn(\mathscr{F}_n)) = n$. For every odd $5 < n < 23$, $rc(Syn(\mathscr{K}_n)) = n$.*

**Proof.** The series of synchronizing automata $\mathscr{F}_n$ guarantees that $rc(Syn(\mathscr{F}_n)) \le n$ for $n > 3$. Arguing by contradiction suppose that $rc(Syn(\mathscr{F}_n)) < n$. It means that there exists some synchronizing DFA $\mathscr{B}$ with less than $n$ states such that $\mathrm{Syn}(\mathscr{B}) = Syn(\mathscr{F}_n)$. Construct its power automaton $\mathcal{P}(\mathscr{B})$ consisting only of reachable subsets. The DFA $\mathcal{P}(\mathscr{B})$ has at most $2^{n-1} - (n-1)$ inequivalent states and recognizes $Syn(\mathscr{F}_n)$. However, by Proposition 14 we have $sc(Syn(\mathscr{F}_n)) = 2^n - n$, a contradiction. Thus $rc(Syn(\mathscr{F}_n)) = n$. The other equality is obtained analogously. □

Another known series of slowly synchronizing automata do not lead to completely reachable series. For example, power automata for $n$-automata $\mathscr{E}_n$ and $\mathscr{E}'_n$ (see Fig. 12) do not contain at least $2^{n-2} - 1$ states. Indeed, any subset containing $\{n-2, n-1\}$ is not reachable in $\mathscr{E}_n$, since $n-2$ does not belong to the image of the state set $\{0, \ldots, n-1\}$ under the action of $a$ and $n-1$ does not belong to the image of the state set $\{0, \ldots, n-1\}$ under the action of $b$. So $sc(\mathrm{Syn}(\mathscr{E}_n)) \le 3 \cdot 2^{n-2} - n + 1$. Analogously, $sc(\mathrm{Syn}(\mathscr{E}_n)) \le 3 \cdot 2^{n-2} - n + 1$. Computer experiments show that these upper bounds seem to be precise values of the state complexity of the languages of reset words for $\mathscr{E}_n$ and $\mathscr{E}'_n$.

Further it would be interesting to study the reset complexity and the state complexity of languages related to slowly synchronizing automata over non-unary alphabet [5]. Finally, we summarize known and obtained in the present paper complexity results related to slowly synchronizing automata in the following table. We denote by $I_n$ the language $I_n = \mathrm{Syn}(\mathscr{A}_n)$. The values of reset thresholds for the corresponding automata are taken from [1].

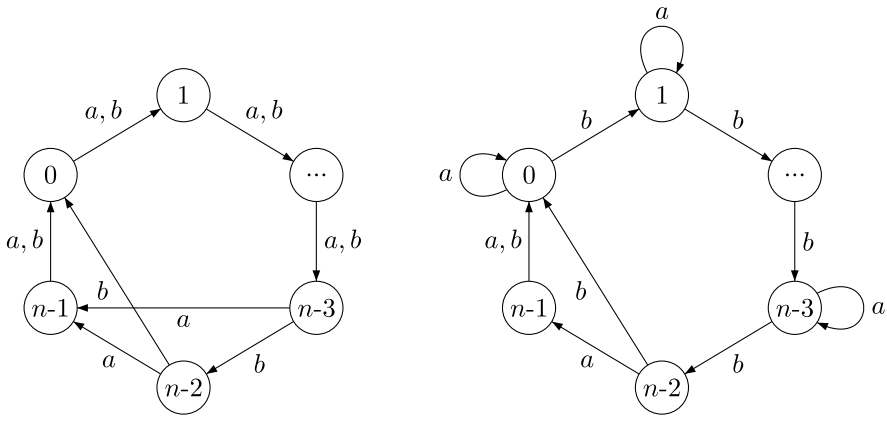


Fig. 12. The automata $\mathscr{E}_n$ (left) and $\mathscr{E}'_n$ (right).

Table 1. Complexity characteristics of languages related to slowly synchronizing automata.

| The DFA $\mathscr{A}_n$ | n | $rt(\mathscr{A}_n)$ | $rc(I_n)$ | $sc(I_n)$ |
|---|---|---|---|---|
| $\mathscr{C}_n$ | $n > 1$ | $(n-1)^2$ | $n$ | $2^n - n$ |
| $\mathscr{L}_n$ | $n > 4$ | $n^2 - 3n + 4$ | $n$ | $2^n - n$ |
| $\mathscr{V}_n$ | $n > 2$ | $n^2 - 3n + 3$ | $n$ | $2^n - n$ |
| $\mathscr{T}_n$ | odd $n > 3$ | $n^2 - 3n + 3$ | $n$ | $2^n - n$ |
| $\mathscr{K}_n$ | odd $5 < n < 23$ | $n^2 - 4n + 7$ | $n$ | $2^n - n$ |
| $\mathscr{E}_n$ | $n > 3$ | $n^2 - 3n + 2$ | less or equal to $n$ | less or equal to $3 \cdot 2^{n-2} - n + 1$ |
| $\mathscr{E}'_n$ | $n > 2$ | $n^2 - 3n + 2$ | less or equal to $n$ | less or equal to $3 \cdot 2^{n-2} - n + 1$ |

## Acknowledgments

The author acknowledges anonymous reviewers for comments and suggestions. Also the author acknowledges support by the Russian Foundation for Basic Research, Grant No. 16-01-00795, the Ministry of Education and Science of the Russian Federation, Project No. 1.3253.2017, and the Competitiveness Enhancement Program of Ural Federal University.

## References

[1] D. S. Ananichev, V. V. Gusev and M. V. Volkov, Slowly Synchronizing Automata and Digraphs, *MFCS 2010*, *LNCS*, eds. P. Hliněný and A. Kučera, Vol. 6281 (Springer, Heidelberg, 2010), pp. 55–65.

[2] E. A. Bondar and M. V. Volkov, Completely reachable automata, *DCFS 2016*, *LNCS*, eds. C. Câmpenu, F. Manea and J. Shallit, Vol. 9777 (Springer, 2016), pp. 1–17.

[3] J. Černý, Poznámka k homogénnym eksperimentom s konečnými automatami, *Mat.-Fyz. Cas. Slovensk. Akad. Vied.* **14**, *[in Slovak]*, (1964), pp. 208–216.

[4] H. Don, The Černý conjecture and 1-contracting automata, *Electr. J. Comb.* **23**(3) (2016) P3.12.

[5] A. Kisielewicz and M. Szykuła, Synchronizing automata with extremal properties, *MFCS* 2015, *LNCS*, eds. G. F. Italiano, G. Pighizzini and D. Sannella, Vol. 9234 (Springer, 2015), pp. 331–343.

[6] D. Kozen, Lower bounds for natural proof systems, *18th Annual Symposium on Foundations of Computer Science*, *IEEE* (New York, 1977), pp. 254–266.

[7] P. Martygin, Computational Complexity of Certain Problems Related to Carefuuly Synchronizing Words for Partial Automata and Directing Words for Nondeterministic Automata, *Theory Comput. Sci.* 2014, **54**(2) ed. F. Ablayev, (Springer, 2014), pp. 293–304.

[8] M. I. Maslennikova, Reset Complexity of Ideal Languages, *arXiv*: 1404.2816 (2012). *Int. Conf. SOFSEM* 2012.

[9] M. Maslennikova, Complexity of checking whether two automata are synchronized by the same language, *DCFS* 2014, *LNCS*, eds. H. Jürgensen, J. Karhumäki, Al. Okhotin, Vol. 8614 (Springer, Heidelberg, 2014), pp. 306–317.

[10] A. R. Meyer and J. F. Michael, Economy of description by automata, grammars, and formal systems, *12th Annual Symposium on Switching and Automata Theory*, IEEE (New York, 1971), pp. 188–191.

[11] F. R. Moore, On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Transactions on Computers*, C-20(10), IEEE (New York, 1971), pp. 1211–1214.

[12] D. Perrin, Finite automata, *Handbook of Theoretical Computer Science*, ed. J. van Leewen (Elsevier, B., 1990), pp. 1–57.

[13] L. J. Stockmeyer and A. R. Meyer, Word problems requiring exponential time, *Proceedings of the 5th Annual ACM Symposium on Theory of Computing STOC* '73, ACM, eds. A. V. Aho (New York, 1973), pp. 1–9.

[14] M. V. Volkov, Synchronizing automata and the Černý conjecture, *LATA* 2008, *LNCS* 5196, eds. C. Martín-Vide, F. Otto and H. Fernau (Springer, Heidelberg, 2005), pp. 11–27.